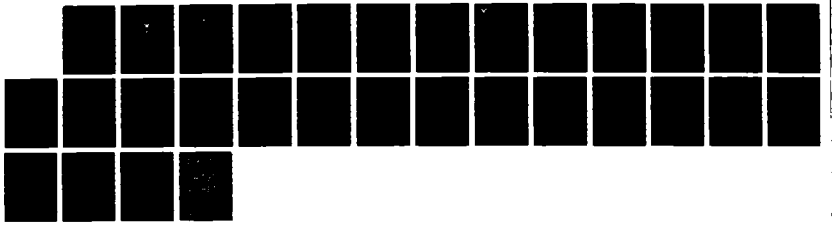
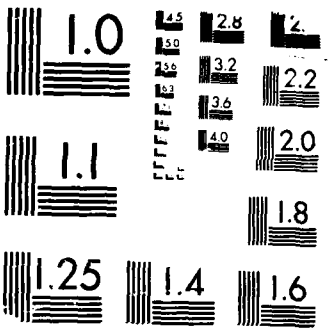
AD-A194 393 WHAT'S ON THE MENU? FRIENDLIER COMPUTERS(U) AIR COMMAND 1/1
AND STAFF COLL MAXWELL AFB AL C A BASILI APR 88
ACSC-88-0235

UNCLASSIFIED F/G 12/5 NL





MICROCOPY RESOLUTION TEST CHART

(2)

AD-A194 393

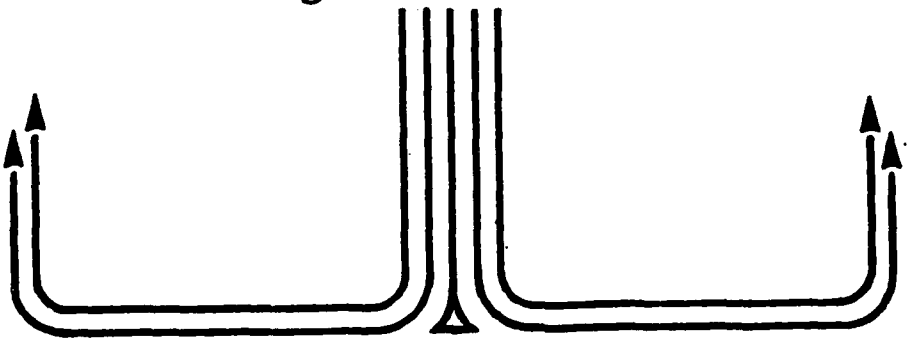
AIR COMMAND AND STAFF COLLEGE

STUDENT REPORT

**WHAT'S ON THE MENU?
FRIENDLIER COMPUTERS!**

MAJOR CARL A. BASILI 88-0235

"insights into tomorrow"



REPORT NUMBER 88-0235

TITLE WHAT'S ON THE MENU? FRIENDLIER COMPUTERS!

AUTHOR(S) MAJOR CARL A. BASILI, USAF

FACULTY ADVISOR CAPTAIN RONALD D. FORD, ACSC/XPO

SPONSOR LT COL KEITH S. BROWN, 1973rd COMM GP/CC

Submitted to the faculty in partial fulfillment of requirements for graduation.

AIR COMMAND AND STAFF COLLEGE
AIR UNIVERSITY
MAXWELL AFB, AL 36112-5542

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

A104 392

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT STATEMENT "A" Approved for public release; Distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 88-0235	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION ACSC/EDC 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Maxwell AFB AL 36112-5542	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

WHAT'S ON THE MENU? FRIENDLIER COMPUTERS!

12. PERSONAL AUTHOR(S)

Basili, Carl A., Major, USAF

13a. TYPE OF REPORT	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
		1988 April	28

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The paper focuses on how to tap the true power, flexibility, and potential of personal computers by making ~~them~~ easier to use. It is a starting point from which computer users can begin to make computers do what they want, when they want, and in a manner they choose. The study includes an evaluation of computer-user interface programs as an alternative solution to learning computer languages and other technical features not directly associated with using computers to accomplish specific tasks. The study concludes that menuing programs are the most powerful interfaces available and offer users ease-of-use and greater productivity.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL ACSC/EDC Maxwell AFB AL 36112-5542	22b. TELEPHONE (Include Area Code) (205) 293-2867 / 22c. OFFICE SYMBOL

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

PREFACE

The following situations all describe a condition that keeps out of reach the great potential computers offer in getting complex tasks done quickly and accurately.

For weeks the boss has been complaining about blank screens. Afterall, he authorized the six million dollar purchase so everyone could have a personal computer on their desk. But now that they're here, hardly anyone is using them. They say the manuals are too technical and they don't have time to learn "computerese."

> How many times have you been asked to set up a system for a novice computer user? Or would you simply like to put your own system together so that you could get to a particular program by pressing a single key? (6:52).
>
> First you get a fresh cup of coffee then make sure there is plenty of paper for the printer. Next you forward all calls to the secretary, sit down at the computer, turn it on and you're ready for two hours of answering prompts to produce the weekly reports. Once again you think, "There has to be a better way!" (16:41).

This paper focuses on how to tap the true power, flexibility, and potential of personal computers by making them easier to use. It is a starting point from which computer users can begin to make computers do what they want, when they want, and in a manner they choose. The author's intent, therefore, has been to evaluate computer-user interface programs as an alternative solution to learning computer languages and other technical features not directly associated with using computers to accomplish specific tasks.

ABOUT THE AUTHOR

Carl A. Basili is currently a Major in the United States Air Force attending Air Command and Staff College, a mid-level Air Force Professional Military Education course, at Maxwell Air Force Base (AFB), AL. He began his military career in 1974 as a graduate of the Air Force's Reserve Officer Training Corps at Pennsylvania State University where he majored in Psychology. After completing technical training at Sheppard AFB, TX (Dec 1974) and Vandenberg AFB, CA (Feb 1975), he served as a Missile Launch Officer in the Titan II ICBM System at Davis-Monthan AFB, AZ. In 1979, he became a Social Actions Officer at Andrews AFB, MD and, while assigned there, also attended the Air Force's Squadron Officers School at Maxwell AFB, AL (Sep 1980). He continued in this career field, serving at Royal Air Force Station Chicksands, England, from 1981 through 1983, and at Los Angeles Air Force Station, CA from 1984 until 1985. During this latter assignment he received his first formal training in personal computer use from Zenith Data Systems, Incorporated. Then, after completing Ground Launch Cruise Missile technical training at Davis-Monthan AFB, AZ (Aug 1985), he served as Senior Flight Commander for the fighting 487th Tactical Missile Wing at Comiso Air Station, Italy, until June 1987 when he returned stateside. In his last two assignments, Major Basili used personal computers to assist in administrative and mission-related operations. It was these experiences that focused his attention on "computer friendliness" as a means of "doing more with less." Major Basili is married, with two children, and his major interest is home computing, although he still finds time to devote to his other interests of classic cars and woodworking.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Part of our College mission is distribution of the students' problem solving products to DoD sponsors and other interested agencies to enhance insight into contemporary, defense related issues. While the College has accepted this product as meeting academic requirements for graduation, the views and opinions expressed or implied are solely those of the author and should not be construed as carrying official sanction.

"insights into tomorrow"

REPORT NUMBER 88-0235

AUTHOR(S) MAJOR CARL A. BASILI, USAF

TITLE WHAT'S ON THE MENU? FRIENDLIER COMPUTERS!

I. Purpose: Determine the advantages and disadvantages of computer-user interface programs that make a computer's Disk Operating System environment easier to use and the characteristics an "ideal" program should contain. Also, demonstrate the usefulness of computer-user interface programs by evaluating one such program against "ideal" program characteristics and draw conclusions based on findings.

II. Discussion of Analysis: Personal computers are tools of fantastic potential, capable of accomplishing almost any task. Yet this capability is seldom achieved. Getting a computer to do what a user wants it to do is often difficult because the computer is a machine, rigid in design and only able to do exactly what it is told to do. Humans, on the otherhand, are flexible and can apply judgment. The computer-user relationship, therefore, is plagued by errors resulting from incompatibilities in the area of communications. Only when this hurdle is overcome, can the true potential of computers be achieved. The biggest obstacle to improving the computer-user interface is the computer's Disk Operating System (DOS); the basic "instruction set" that allows the computer to do something, anything, and all things. The use of programs specifically designed to accomplish this task provide an interface for people such that computers really do become tools to help them in their work. This paper specifically defines what these programs can do, discusses how to select the right program based on needs of the using population,

CONTINUED

cites the advantages and disadvantages of using computer-user interface programs (CUIP), determines characteristics of the "ideal" program, evaluates one commercially available program against "ideal" characteristics, and concludes use of such programs do make computers easier to use.

III. Findings: The key to easier-to-use computers involves taking user needs into account. Since users each have different requirements and levels of experience, the basic obstacle common to all is the computer's DOS. CUIPs solve this problem by isolating users from the DOS. Of the many types of CUIPs, menuing programs are the most powerful, offering features and capabilities individuals and organizations alike desire and need. Isolating users from the DOS, however, can mean they may never become familiar with the basic instruction set of the computer. Still, if computer output, that is a product, is all that's of interest, then users need never learn about the DOS, thus making this sole disadvantage unimportant. Nevertheless, the "ideal" menu should take this into account by offering the user the choice to use the menu or not. This characteristic, as well as many others discussed in the paper, make up a basic "shopping list" of critical features the "ideal" menu should have. Selection of any menuing program should then include these features plus as many user-desired, though not essential, features. Finding the right menu involves evaluating those available, not only against a list of "ideal" and desired features, but based on current and future needs to preclude early obsolescence.

IV. Conclusions: Enhancing computer productivity is the whole idea behind making computers easier to use. Choosing the right menu with the right features will make computing easier and enhance productivity. Additionally, the tenets of menuing programs that accomplish this end may be applied to application programs (such as word processors) as well. "Friendly", easy-to-use programs allow users to produce sooner, faster, and with greater accuracy.

Chapter 1

INTRODUCTION

This chapter introduces the idea of making small or personal computers easier to use. It contains a discussion of what constitutes computer "friendliness," identifies who can benefit from friendlier computers, and provides a definition of what computer-user interface programs (CUIP) are and what they promise in the way of friendliness. Let's begin with the "what."

WHAT IS "USER FRIENDLY"?

Friendlier computers and user-friendliness are both terms that have been around for quite some time, even though a clear definition has yet to be developed. Both connote a state that is usually considered desirable although no specific criteria exists. Computer friendliness is best defined as user-sensitive. Henry Simpson, a writer and computer programmer, puts it this way: "In designing anything that people will use, it is a good idea to find out as much about the users as possible. That way you can tailor your design to fit their needs" (3:2).

Accounting for user needs, then, is perhaps our best method of describing "friendliness", especially when we're talking about small computers in the home or office. The next question is "Who are the users and what do they need?"

WHO ARE THE "USERS"?

Users are more than those who directly interact with a computer. They are people who have varying levels of expertise and different requirements for using computers. Users can also be entire organizations or single offices. Individual users can be grouped into three broad categories. First, there are the "novices". The novice user is usually a person with little, if any, background in computing but understands how to use a particular application program (such as a word processor). Next, there's the "experienced" user. This person understands the basics of how computers operate and can use particular application programs as well as interface directly with the computer. Finally, there's the "advanced" user. This person (such as a programmer) has a thorough understanding of computers and can do

just about anything with them. Organizations, when viewed as users, are merely groups of people (a few to several hundred) from all the above categories. Having now defined who the users are, and remembering that computer friendliness involves taking the user into account, we can begin to surmise user needs.

WHAT DO USERS NEED?

According to H. T. Smith, a psychologist and computer consultant,

> Communicating with a computer is typically a very unsatisfying experience. Most of the communication usually consists of exchanges involving short, serially ordered strings of commands or messages. The onus is on the users to check that their inputs. . . are meaningful and. . . correct (1:14).

Herein lies the argument for friendlier computers based on individual user needs. Smith puts this premise into focus by writing: " While the programmer may find fascination, the person just introduced to a computer for the first time may exhibit a markedly different set of emotions based on fear, awe and general uncertainty" (1:13).

This being the case, one might assume the more experienced a user, the less the importance of "friendliness" becomes. However, getting a computer to do what you want can cause problems for almost anyone, regardless of experience level. This is because:

> Before a computer will run. . . [a program], you need a special [program] called a disk operating system, or DOS for short. . . . the DOS makes the machine practicable by loading into memory. . . various routines that perform two general 'housekeeping' functions. First, DOS gives you control over operations by letting you load and run your application programs. Second, it provides a special set of commands for copying files, deleting files,. . . and so on. . . . As significant as DOS is in operating a [computer], it ranks low on the friendly scale and requires a fair amount of 'sweat equity' to master (4:52).

Thus, the DOS becomes our primary route to friendlier computing. When using a DOS on a day-to-day basis, there are many occasions when a user must enter several separate commands, one after the other. Sometimes, users must type these commands with about a half-dozen hard-to-remember parameters several times throughout the working day. Typing these commands in manually can cause many problems. A required command may be mistyped,

forgotten, used out of sequence, or may be beyond the understanding of the novice user. Wouldn't it be nice to be able to combine several commands into a single command, or have a list of commands on the screen requiring the user to merely press a single key, freeing him or her from the computer's keyboard and letting them go on with more important things rather than redundantly entering commands (7:40)? Guiding the user through a complex task without error sounds like a good place to begin the effort to make computers friendlier.

HOW CUIPs HELP

Enter computer-user interface programs, or CUIP for short. These programs offer to make computers friendlier by isolating users from the DOS and providing single keystrokes to perform all the necessary functions to execute particular applications or perform specific operations. They create a single environment in which users select various applications from a main menu. With just a keystroke or two, users load programs and enter any application without having to remember specific commands. For novice users, CUIPs offer instant access to needed applications. For experienced and advanced users, they offer quick and easy execution of complex applications or operations. Friendliness through an easy-to-use program, single keystrokes versus hard-to-remember commands, and elimination of redundant keyboard entry -- that's what CUIPs offer and how they propose making computers friendlier to novice and experienced users alike. But like users, there are different types of CUIPs, each offering different levels of friendliness, capabilities, and features. The task now becomes matching CUIP capabilities to individual users and their specific needs -- the topic of the next chapter.

Chapter 2

DO WE REALLY NEED CUIPs?

Chapter One explained the key to friendlier computers lies in taking user needs into account, that users (individuals and groups) have different needs based on experience level and requirements, that a computer's DOS is the basic obstacle in getting a computer to do what the user wants, and that CUIPs promise to overcome this obstacle by isolating users from the DOS. If the answer's this simple, why doesn't everybody use a CUIP? Many individual users do (15:--). But to understand this and determine if there's a need for CUIPs in a particular organization, CUIPs first need to be categorized by the features they offer and matched to organizations (different groups of users) based on specific needs.

MENUS, SHELLS, MACROS

CUIPs, like users, can be broken down into three broad categories: menus, shells, and macros. Menus are the most sophisticated form and provide the user with a list of applications and operations which can be selected by a single keystroke. They come with a variety of features, prewritten menus, and facilities to create custom menus to replace or supplement those provided. Shells offer ". . . a menu system of DOS commands, programs,. . . [but]. . . don't let you build custom menus." Macros are merely programs which allow the user to assign several keystrokes or commands to a few keys. There are no menus so users must remember what functions are assigned to which keys (4:53). Matching these CUIP types and capabilities to user needs becomes the next step toward friendlier computers.

NEEDS AND CAPABILITIES

Since novice users may have little, if any, familiarity with the DOS, they would probably benefit most from menus. Likewise, experienced users don't need the sophistication offered by menus and might best be served by shells. Finally, macros are the best choice for advanced users, such as programmers, since their skills eliminate the need for menus altogether.

Each of these user groups interact very differently with computers. But users within each group also have different needs. For example, a menu which offers six selections and then asks the user to give a variety of different information for the application selected is helpful to the novice. But soon the novice may find this detailed interaction irritating as they become more experienced. Additionally, one individual that is familiar with DOS operations may be a novice when it comes to a particular application, such as word processing. This is often the case in organizations. Therefore, the issue is not matching user types to CUIP types but, rather, matching user needs with CUIP capabilities.

If only individual users are considered, then a direct "user-need-versus-CUIP-type" comparison may be appropriate. "If, for example,. . . you plan to do nothing but a few basic DOS operations such as. . . file copying. . . and run a small number of programs, then you'll need no more than a simple shell with a limited number of features" (4:53). Users who are experienced with the DOS and merely want more control over DOS operations, or to save time telling the computer what to do, will probably find macros more than adequate. It's only when organizations are considered as users that all benefits offered by CUIPs are seen.

Since organizations are made up of users of different experience levels and the organization itself may have specific needs (such as security), selecting a CUIP can be very complicated. "The majority of working people using [small computers] are clerical, although the trend for the future heavy use is middle management" (14:73). This being the case, the need increases to use computers to get the job done faster, without error and thereby improve productivity -- the original purpose for computers in the first place. But individual users, regardless of experience level, want more that just ". . . sheer speed or accuracy; people want computers to be easy, and they want computers to know how to do everything" (14:73). Organizations, on the other hand, have needs, too, for security and ease of use across the user experience spectrum. For example, ". . . suppose you're the boss and you want your employees to have access only to the programs for which they're responsible. This requires a fairly elaborate [CUIP] with custom-menuing, security, and, perhaps, an 'audit trail' [log of system users]. . . " (4:53). But users aren't the only ones concerned with friendliness and ease of use. This need is also a primary focus and concern of the industry that writes the programs users use.

THE INDUSTRY'S ANSWER

The need for friendlier computers can be better understood by listening to the comments of the computer industry's leaders. They were asked "What will make computing easier or more diffi-

cult for business users over the next year?" Here's what they said (15:287,289,340):

> Phil White, President of Wyse Technology -- "Technological advancement, without some insight into the needs of the user, makes computing increasingly complicated. We've got to focus on the needs of the user. . . if the personal computer is to remain a tool for the nontechnical person."
>
> Tom Yuen, Executive Vice President of AST Research -- "Common user interfaces. . . will make computing easier . . . by giving users a more familiar set of screen and keystroke conventions that cut across diverse computing hardware and application-software products."
>
> Philippe Kahn, President of Borland International -- "The emergence of more flexible, programmable software that offers a better 'user environment' and more performance is likely to bring more users into the computing environment and to encourage all users to try more and to do more with their computers."

The industry has considered the user, identified what it believes to be a valid need, and responded with CUIPs we termed earlier as menus. Menus today promise users, both organizations and individuals, features most desired and needed by all, and in one package. But while menus offer what appears to be the ideal solution, there are some drawbacks in using them. The next chapter looks more closely at this issue through the pros and cons of organizations adopting menus as the means for developing friendlier computers and achieving increased productivity.

Chapter 3

THE GOOD, THE BAD, AND. . . ?

The first two chapters portrayed menus to be the most powerful of CUIPs, offering features both individuals and organizations desire and need most. But there is some debate on whether or not organizations should adopt and thereby become dependent on menus to increase user productivity. The argument against using menus, as well as all other types of CUIPs, centers on their very purpose; making computers easier to use by isolating users from the DOS. To decide if a user, whether an individual or organization, should adopt a menu, careful consideration must be given to this very subject. Examining the user's needs or purpose for using a menu is a must before "jumping on the bandwagon." And the best way to do this is to look at what can be gained versus what may be sacrificed in light of the user's needs.

ADVANTAGES

As discussed earlier, the logical argument for using a menu in the first place goes something like this:

> Computers were designed to save time and effort in accomplishing particular tasks. But the biggest drawback to using a computer is in coping with its DOS. So to increase productivity, simply isolate users from the computer's DOS by using an easy-to-understand communications interface.

This is just what menus do. They successfully isolate users from the complex details of the DOS by handling DOS housekeeping functions just as secretaries "often isolate their bosses from the distractions of phone calls by handling the calls themselves" (4:52). Specifically, menus increase productivity by allowing a single keystroke to accomplish a task in place of complex DOS commands and by providing security, accuracy and a variety of other features a particular user needs. These features usually cannot be easily achieved or are not offered by the computer's DOS. Here, then, are the basic gains to be made by adopting a menu. But, what's lost or sacrificed in the process?

DISADVANTAGES

The argument against using a menu, as stated earlier, centers on isolating users from the DOS. Isolation, for the purpose of the argument, translates to never knowing or learning how to use the DOS. John C. Dvorak, contributing editor to PC Magazine, sums it up this way (5:65):

> *There is a lot of junk available for the user of today's microcomputer. By junk, I mean useless software. . . . the companies that make these products all hope to crash in on the same market: the naive user, the bonehead. . . . [one] product in this group is the so-called [menu], a program that takes over the screen you would normally see. . . . Instead of typing [a DOS command] you scroll around a huge menu of files and highlight [one of them]. Then you hit the Return key, a function key, or some letter on the keyboard and the program executes. It seems much simpler just to type [a DOS command]. . . . menuing programs are designed to be used by a corporate guy who is setting up a limited number of applications for use by some department. The assumption of this software is that users will run only a few specific applications, ever. The typical "approved" menu comes up when the machine is turned on and it says: "Do You Want To Do: (A) Word Processing, (B) Lotus, (C) Print the Directory of Files, (D) Quit." For this the corporate guy gets $75,000 a year. . . . My advice: buy a book.*

THE GREAT DEBATE

Mr. Dvorak's argument points out limitations in some menuing programs. Limitations of flexibility, capability to handle only a few operations, and lack of features. But his advice to buy a book really gets to the heart of his the matter; users should learn the computer's DOS. And many agree with him. Here's what some other computer users said (8:17):

> S. Hamill Horne: "These programs [menus], like bicycle training wheels, only hinder a user's speed and mobility on the keyboard."
>
> Boyd Wood: "Our biggest problem in supporting our customers is dealing with their [menus]. It seems nobody knows what 'C>' means."

But while some agree with this school of thought, others, including the computer industry, do not. What they say focuses the argument even further (8:18).

Michael G. Reed: "John C. Dvorak is right -- there is a lot of software junk out there. But to discount all [menus] for all people is to forget the several *hundred* years he's spent becoming familiar with those obscure and often illogical DOS commands."

Robert A. Burns: "I use a [menu], and it certainly makes my life a lot easier. Sure I know enough about DOS to find my way around a hard disk, and I can type long, cumbersome DOS commands as well as anyone. But why should I go through all that trouble?"

The debate still goes on and it continues to be based on user needs as depicted in the preceding quotes. What is gained is ease of use. What is sacrificed is familiarity with the DOS. So, the answer to the question "Do we really need a menu?" is yet another question: Does the user need to be familiar with the DOS or just have access to DOS operations and particular application programs, or both? The answer is both. No program can be "all things to all users," but if a *menuing program* offers the capability for the user to decide to use it and to what extent, or not to use it at all, then the argument becomes elementary. If users don't like a *menu* with this on-off feature, they won't use it because they make the decision, not someone else such as "the corporate guy."

We've now looked at what is to be gained and sacrificed in light of user needs and determined that a menuing program, to truly enhance productivity, must be flexible enough for all users of a particular computer. Here, flexibility refers to users deciding whether or not to use a menu. This is a feature any menu adopted for use should have. But it's not the only important feature, as we'll see in the next chapter which looks at characteristics of the "ideal" menu.

Chapter 4

THE "IDEAL" MENU!

Menus are merely a collection of features, or program steps, that accomplish specific functions or tasks, like copying files. And since menus are the most powerful CUIPs available, promising the greatest ease of use, they are laden with user-friendly features. Some features are critically important, such as the on-off feature discussed in the previous chapter, while others are not. At a minimum then, the "ideal" menuing program should meet as many user needs and desires as possible by incorporating all the critically important features. This chapter reviews some of the better features found in various menuing programs. Below is a list of these features, but only those considered absolutely necessary are mentioned. Included is a brief description of what each feature does and its relative importance to users.

HARDWARE AND SOFTWARE COMPATIBILITY

This is the most important consideration. Will the menuing program properly work with the hardware (i.e., computer system) and software (i.e., application programs) already in use? Always try to check compatibility first. This is most often done by reviewing the documentation that comes with the menu and software reviews found in many popular microcomputer periodicals. It takes time to do this, but a few hours spent here may save valuable time and avert disaster later.

DOCUMENTATION

Be certain to get all the instructions or documentation that goes with the menu. This is important not only to ensure compatibility but to aid the user if problems arise. In fact, if a menu comes without documentation or has poorly written documentation (i.e., incomplete, hard-to-understand) then don't select that menu for use.

OPERATING MODES

There are two modes of operation: memory resident and non-memory resident. Memory resident refers to menus which remain in the computer's memory, taking up space, while a particular operation or application is running. While a computer's usable memory is limited, and this mode limits further the amount of memory available to do a task, the advantage is speed. For simple operations such as copying files and formatting disks, this is very desirable. However, when using an application program, such as a word processor, the application may need all the computer's available memory. In this case, non-resident mode is desirable. This mode allows the menu to turn over control to the application program and free up computer memory the menu once occupied. The ideal menu should have both these modes of operation for maximum flexibility (13:275).

CUSTOMIZATION AND HUMAN FACTORS DESIGN

These terms refer to the way a particular menu appears on the computer screen (i.e., how it looks) and how it communicates information to the user. Several design considerations are involved in making a menu user friendly. Design features of color, format, location of menu selections on the screen, size, organization of information, and a variety of other, less important features that make up the menu's display on the screen often determine if a menu will be used or not (12:--). The ideal menu must allow user-defined customization of menu screens that are logically organized, communicate in plain English, and offer users as many special features (i.e., colors, clock display, etc.) as possible.

DEGREE OF USER CONTROL

This feature is based on the premise that detailed menus are very helpful to the novice user but become a burden as the user gains experience. User control (i.e., on-off feature) was touched on in the last chapter as a necessary feature for any menu adopted for organizational use where people of different experience levels using the same computer have different individual needs. But the ideal menu goes far beyond simple on-off control. It should permit a full range of control from totally manual (i.e., menu off) to fully automatic (i.e., by a prescribed procedure). This spectrum of control includes automatic control with switching to manual control, that is, the user monitors menu operations and may take control or interrupt the menu at any given time. This flexibility in control permits the ideal menu from becoming obsolete, affords new users within an organization to operate a computer quickly with little supervisory assistance, and precludes penalizing more experienced users (1:49).

PROGRAM FEEDBACK

The ideal menu should always let the user know it is "listening." If a user makes a mistake, the program should respond with something like "You pressed the wrong key, Please try again." If a menu operation takes time to do, the program should respond with a status report and ask the user to wait. Feedback to the user is essential in preventing errors made out of frustration, unfamiliarity, or impatience (9:--).

ERROR HANDLING

This feature attempts to deal with the issue of how much help a user who makes a mistake can be given. As with program feedback, the ideal menu should prevent anything from happening if an incorrect key is pressed and at the same time tell the user what has happened and what to do next. If a particular operation, such as formatting a disk, precludes preventing a mistake, the ideal menu should provide a warning message and a way out, like "This selection will format the disk in Drive 'A' and all data on the disk will be lost. Press any key to continue or press the 'ESC' key to abort the operation and return to the menu." This example is lengthy but typical of menu feedback and error handling for novice users. Experienced users will find a less detailed message more appealing. Error handling, therefore, is dependent on user experience and the ideal menu should provide maximum flexibility in designing controls and messages (2:35).

OVERALL FLEXIBILITY

The term flexibility keeps coming up throughout this chapter because meeting user needs, our primary objective, means being flexible. The ideal menu "must be easy to modify. . . . However good the [menu] design, it can never be 100% what [all] users need if only because needs continually change, and what is 100% at design time will be something less when the [menu] is brought into use" (2:16).

SIMPLICITY AND CONSISTENCY

The ideal menu should support these two principles. Simplicity refers to the layout of how the menu and choices are displayed. If the computer screen is cluttered and complex, users are more likely to make errors or not use the menu at all. This concept was previously discussed under customization and human design factors. Consistency, on the otherhand, refers to reducing the number of things that must be learned by "performing similar tasks in similar ways (3:112). For example, if a menu is made up of several selections each leading to another sub-menu

that also has several selections but a different format, users may become confused. Coupling simplicity and consistency to minimize the scope of learning required to use a menu precludes user confusion and speeds the user on the way to faster results.

SECURITY AND PASSWORD PROTECTION

Chapter Two mentioned this feature as a way to prevent unauthorized access to particular files on a computer used by several people. Another use of this feature overlaps with error handling described above. A password can be used to prevent users from making critical errors, like erasing important files. For example, if one menu selection involves formatting a disk, a request to enter a password before proceeding would stop someone from accidentally erasing files through an inadvertent format operation.

This "shopping list" of features is generic and should be applicable no matter what equipment is used. However, the list should not be considered all inclusive. Some users may have specific requirements not mentioned here, and then there are highly desirable or just "nice-to-have" features as well. But in general, these features should be considered the minimum acceptable when selecting any menuing program where users with different experience levels use the same computer. Using this list as a basis for selection, our attention can now be directed at evaluating menuing programs, the topic of the next chapter.

Chapter 5

AND THE WINNER IS. . . ?

Selecting a menuing program that meets as many user needs as possible and still conforms to the minimum acceptable list of features discussed in the last chapter can be a long and tedious process. This task can be significantly reduced by reviewing what others have already said about some of the computer industry's leading menuing programs. A comparison of several programs based on software reviews in current microcomputing periodicals can reduce the choices to only a few. Then a further comparison against "ideal" menu features outlined in the previous chapter will narrow the choices even further. But, rather than compare several commercially-available menuing programs, this chapter provides a concrete example of the evaluation process by examining only one program. Individual users may then accomplish their own reviews, adding their own unique needs to the list of critical features, and select a menu that's right for them.

Since no one menu can be all things to all users there can be no clear winner. However, for the purpose of this review, one such program, AUTOMENU by Magee Enterprises, was selected based on its myriad of user friendly features. A look at what other computer users have said begins the evaluation process. According to Joe Pierce, technical assistant to Soft Sector Magazine (11:63),

> AUTOMENU is much more than just another [CUIP]. It is a complete user-definable working environment. . . . AUTOMENU seems well-suited for the corporate environment. A micro-manager could design a menu system and standardize it across an organization. . . . Overall, [AUTOMENU] is a slick and well-designed product. It performs well, and looks like the perfect product for those who have trouble getting the hang of, or do not like [DOS].

Specific features praised in this review were customization flexibility, password protection, excellent documentation, online help (i.e., explanatory messages contained in the program that can be viewed by the press of a key),and hardware compatibility with a wide variety of MS-DOS machines (e.g., IBM personal computers and similar types). There were no disadvantages or shortcomings cited (11:63).

Another, very detailed review of AUTOMENU by Hal Nieburg, contributing editor to Computer Shopper, cites the program as (10:471):

> . . . both elegant in convenience and powerful in its array of functions. It is a well conceived tool of considerable flexibility and logical clarity. . . offering not only normal DOS functions, but other opportunities as well, enough to boggle the mind. . . . a major piece of software, comprehensive and well conceived, embodying wrinkles that have taken a long period of development.

This review is very extensive and covers in detail what can be gained and sacrificed when opting for AUTOMENU as the menu of choice. From Mr. Nieburg's review, however, AUTOMENU's successes appears to far outweigh its shortcomings, or rather, its limitations.

> AUTOMENU was designed to enable users to organize and control the entire micro environment. You can create menus that access your application programs, and perform DOS commands with a single keystroke. You no longer have to remember complex DOS commands, syntax or subdirectories, because the program will do it for you. You can go beyond DOS. . . (10:472).

AUTOMENU's features that contributed to this very positive review included extensive customization capabilities; compatibility with most application programs; thorough and complete documentation; memory resident and non-resident operation modes; hardware compatibility and support for all types of computers, printers, monitors, etc.; user-defined input and output in plain English (i.e., program feedback, error handling, and menu appearance); password protection; flexibility of user control covering the entire spectrum of manual to automatic operation; and on-line user help.

As stated, this is a detailed review covering both good and bad points of the program. The bad points are really limitations versus disadvantages. These limitations include a lack of graphics and icons (i.e., pictorial representations of menu selections versus plain English text) and difficult, complex network support (i.e., using the program when several computers are connected together and all share information and programs). While these limitations exist, they are probably not of importance to most users (10:--).

From both of these reviews, it's easy to see that AUTOMENU is a very powerful and extensive menuing program that can satisfy almost all users. AUTOMENU's flexibility is the key in appealing to a variety of users having different experience levels, while

its plethora of features offer users with different needs the power and capabilities most desire. In fact, Magee Enterprises, the program's author, claims AUTOMENU has been adopted and is currently used by half the Fortune 500 companies; the nations's top 500 companies judged on their success and annual profits (17:1). Impressive credentials, but how does this "super" menu stand up to the "ideal" menu?

AUTOMENU stands up very well when compared to "ideal" menu features described in the last chapter. To recap, "ideal" menu features included hardware and software compatibility, thorough documentation, both memory resident and non-resident operating modes, customization flexibility and a human factors design, total user control from manual through automatic, program feedback, error handling, overall flexibility that permits easy modification, simplicity and consistency, and password protection. AUTOMENU has each of these features and many more, however, there are some other limitations not mentioned in the above reviews.

These limitations are minor, but deserve mention nonetheless. Each limitation discussed below is a result of the author's testing and use of AUTOMENU over a six month period. "Ideal" menu features not covered are considered satisfactory implementations.

DOCUMENTATION

While AUTOMENU incorporates all "ideal" menu features, some are limited. The first is the documentation. While it is clear, it is also somewhat technical and lengthy (62 pages). This is more than a novice user can be expected to use or understand. Therefore, an experienced user must design and customize menus to include messages for novice users on what to do within each menu so as to avoid confusion.

OPERATING MODES

AUTOMENU's memory resident operation mode is also limited. It can only be used when other memory resident programs are not in use, otherwise conflicts occur that result in lost data and cryptic error messages. Particular memory resident programs to avoid using with AUTOMENU's resident mode are those that change screen color (unnecessary since the program does this) and keyboard utilities that permit reassignment of key meanings (i.e., macro programs discussed in Chapter Two).

PROGRAM FEEDBACK AND ERROR HANDLING

Limitations in these areas are not the result of the program, but instead occur when the user designing a menu fails to imple-

ment these features. Since the program's communication or output to the user is user-defined, care and forethought must be used when creating menus. Program feedback and error handling messages are not automatically generated, but displayed only when the experienced user creating a menu remembers to incorporate these type messages. This is a necessary trade-off to ensure such messages are user-defined, however, a prompt by the program during menu creation would be nice.

These limitations, along with those described in Nieburg's review, are only minor detractors from an otherwise sophisticated and powerful program. Total flexibility can be achieved in menu operation and appearance. Users, regardless of experience level, will find AUTOMENU useful, helpful, and very friendly. In terms of features, the program includes more than most other similar type programs reviewed by the author. This is important for the organization adopting a menu for a population of users with a wide range of needs. While no one user will use all of AUTOMENU's features, all its features will be used by such an organization of diverse users. This is because AUTOMENU is not just a menu, but a menuing system that can create menus of fantastic variety for user's with diverse needs.

As if this isn't enough, there are still more benefits AUTOMENU offers. One is its cost. It's very cheap, in fact, it's free. AUTOMENU is a shareware program. Shareware means that, while the program is copyrighted, it may be copied, distributed, and used freely so long as no payment or commercial benefit (i.e., profit) is received. A copy of AUTOMENU, including documentation, may be obtained through non-profit user groups and a variety of Bulletin Board Systems (i.e., computers connected to each other via telephone). Also, if users wish to be registered owners of the program, they may contact Magee Enterprises at 6577 Peachtree Industrial Boulevard, Norcross, Georgia 30092-3796 or call 404-446-6611. Registration costs $50.00 and entitles owners to a current copy of AUTOMENU on disk, a printed manual, notification and copies of program updates, and technical support.

This concludes the evaluation process. Users should similarly evaluate other menuing programs. After comparing several, and determining what features are offered by each, look at the future before making a final selection. Remember, selecting a menuing program with a limited number of features may meet immediate needs, but does not preclude future obsolescence. The best advice is to choose a menu that has the most features, even if all aren't used immediately. But, if time is short, and evaluation of several programs isn't feasible, AUTOMENU will probably fill the gap. Now where to from here? On to the next chapter, of course, and a brief recap of the key points in this study plus some food for thought.

Chapter 6

NOW WHAT?

With the evaluation process complete, it's time to "get on the bandwagon." This doesn't necessarily mean using a menu, however. What it does suggest is making computing easier, faster, and production-oriented rather than learning-oriented. Enhancing productivity is the whole idea of friendlier computing, and if a menu succeeds in accomplishing this, then adopt a menu. But menus only represent one method. The thrust of this study and the characteristics describing friendliness revolve around user needs which can be applied not only to deal with a computer's DOS, but to individual application programs as well. Features which make a menu useful to users also represent a guide for individual applications (e.g., word processing) that can make them easier to use and boost productivity even further. To better understand this idea, a brief review is in order.

LOOKING BACK

The main theme of this paper has been to make computers easier to use with the objective of enhancing user productivity. To achieve this end, we determined user-friendliness involves taking user needs and desires into account. For humans to easily use computers, it's better to develop systems that communicate on human terms rather than demand humans communicate on a computer's terms. The first obstacle to overcome, then, became the computer's DOS, a cryptic, sometimes illogical control program that tells the computer what and how to do something.

The computer industry offered a solution to this problem with a variety of CUIPs designed to isolate users from the DOS and create a more flexible, logical, and easily understood communications interface. The most powerful of these are menus that incorporate features individual users and organizations need and desire most.

But because menus isolate users from the DOS they may never learn to use the DOS and become dependent on menus for all computer operations. This drawback has stopped many users from adopting menus for productivity enhancement. However, if productivity is the desired end rather than learning to use the DOS, then this disadvantage is of minor importance. This is particu-

larly the case when considering organizations (composed of individual users with different experience levels) as users.

With this argument aside, seeking the most flexible menuing program offering the most features becomes the key starting point for increasing productivity within organizations. The ideal menuing program should provide a user-defined, highly flexible structure that offers several, absolutely necessary features to ensure ease of use.

With these features identified, a simple evaluation process permits user selection of menus that meet ideal standards and user requirements. The process involves reviewing what others have already said about particular menuing programs users may find interesting, comparing these menuing programs with the "ideal" and narrowing the choices to as few as possible with the "tie-breaker" being specific features required by the organization and desired by most users.

TODAY AND TOMORROW

Choosing the right menu with the right features will make computing easier and enhance productivity. But why stop there? As suggested at the beginning of this chapter, apply user-friendly criteria when selecting application programs, too. Just as a menu isolates a user from the DOS, a menu-driven application program can save time and effort and help users to quickly learn to use it, thus becoming productive much faster. This, then, is the challenge. Instead of learning a cryptic language in order to use a computer or an application, cut out the learning step. Look for programs that do the work. Easy to use programs, whether DOS menus or applications, permit users to produce sooner, faster, and more accurately!

BIBLIOGRAPHY

A. REFERENCES CITED

Books

1. Arblaster, A., et al. Human Interaction With Computers. New York: Academic Press Incorporated, 1980.

2. Mehlmann, Marilyn. When People Use Computers. Englewood Cliffs, New Jersey: Prentice-Hall Incorporated, 1981.

3. Simpson, Henry. Serious Programming for the IBM PC/XT/AT. Blue Ridge Summit, Pennsylvania: Tab Books Incorporated, 1985.

Articles and Periodicals

4. Beechhold, Henry F. "Getting More From PC-DOS and MS-DOS." Family and Home-Office Computing, Vol. 5, No. 10, October 1987, pp. 51-53.

5. Dvorak, John C. "Sad Cures For User Idiocy." PC Magazine, Vol. 6, No. 14, August 1987, pp. 65.

6. Hogan, Thom. "May I Please See a Menu?" Business Software, Vol. 3, No. 12, December 1985, pp. 52-58.

7. Humphress, Danny. "A Batch of Commands." Soft Sector, Vol. I, No. 10, May 1985, pp. 40-42.

8. "Letters to PC Magazine: Shell-Shocked." PC Magazine, Vol. 6, No. 18, October 27, 1987, pp. 17-18.

9. Margolis, Neal. "High User Overhead Mars Interface Design." Computerworld, Vol. XX, No. 17, April 28, 1986, pp. 65-72.

10. Nieburg, Hal. "Automenu, Shareware to Design Your Own Shell." Computer Shopper, Vol. 8, No. 1, January 1988, pp. 471-472.

11. Pierce, Joe. "Automenu - Not Just Another DOS Shell." Soft Sector, Vol. IV, No. 8, March 1987, pp. 63.

CONTINUED

12. Reilly, Susan S. and John W. Roach. "Designing Human/Computer Interfaces: A comparison of Human Factors and Graphic Arts Principles." Educational Technology, Vol. XXVI, No. 1, January 1986, pp. 36-40.

13. Rowinsky, Walt. "Having It All - DOS Shells With Macros." PC Magazine, Vol. 6, No. 16, September 29, 1987, pp. 275-290.

14. Smart, George M. Jr. "The Big Picture." Soft Sector, Vol. III, No. 7, February 1987, pp. 73-75.

15. "View From the Top." Personal Computing, Vol. II, No. 10, October 1987, pp. 285-340.

16. Yearwood, James G. "Super Batch Takes Monotony Out of Frequent Application Use." Soft Sector, Vol. II, No. 11, June 1986, pp. 41-42.

Unpublished Materials

17. Magee Enterprises. "Automenu Software Management System Version 4.0." Users' Manual. Norcross, Georgia: Magee Enterprises, May 1987.

B. RELATED SOURCES

Books

Angermeyer, John, et al. Tricks of the MS-DOS Masters. Indianapolis, Indiana: Howard W. Sams and Company Incorporated, 1987.

Barden, William Jr. What Do You Do After You Plug It In? Indianapolis, Indiana: Howard W. Sams and Company Incorporated, 1983.

Sawusch, Mark. 1001 Things To Do With Your Personal Computer. Blue Ridge Summit, Pennsylvania: Tab Books Incorporated, 1980.

END

DATE

FILMED

8-88

DTIC